Datalog初步理解

主讲人：张 奕 裕　指导老师：左 志 强

目 录

Datalog基础介绍

1. Datalog基础介绍

Basic introduction of Datalog

01 Datalog是一种声明性语义的递归查询语言

- Datalog是一个规范

02 Datalog在近几年迎来了一次复兴

- 在80年代和90年代初得到快速发展
- Datalog在新一代的应用程序中得到复兴

03 Datalog具有语法简单和语义良好的特点

- Datalog提供声明式的高层次编程接口

1. Datalog基础介绍

Basic introduction of Datalog

Operator

$$P_i(t_1, t_2, \dots, t_n)$$

rule

$$P_0 \overbrace{:-}\underbrace{P_1, P_2, \cdots, P_i, \cdots, P_n}_{\text{Body}}$$

Head

Predicate

Terms: variables, constants or functions

- Extensional Database Predicates(EDB)
- Intensional Database Predicates(IDB)

1. Datalog基础介绍

Basic introduction of Datalog

Examples

EDB(input)

edge(a , b).
edge(b , c).
edge(c , d).
edge(d , a).

IDB(output)

r_1: path(X , Y) :- edge(X , Y).
r_2: path(X , Y) :- path(X , Z), edge(Z , Y).

Evaluation

2. Evaluation

1. Bottom-up Evaluation
2. Top-down Evaluation
3. Magic sets

2.1 Bottom-up Evaluation

01 Naive Evaluation

```
edge(a , b).
edge(b , c).
edge(c , c).
edge(c , d).
```

```
path(X , Y) :- edge(X , Y).
path(X , Y) :- path(X , Z), edge(Z , Y).
```

$I =$

path

X	Y

,

edge

X	Y
a	b
b	c
c	c
c	d

$T_p^1(I) =$

a	b
b	c
c	c
c	d

,

a	b
b	c
c	c
c	d

$T_p^2(I) =$

a	b
b	c
c	c
c	d
a	c
b	d

,

a	b
b	c
c	c
c	d

$T_p^3(I) =$

a	b
b	c
c	c
c	d
a	c
b	d
a	d

,

a	b
b	c
c	c
c	d

$T_p^4(I) =$

a	b
b	c
c	c
c	d
a	c
b	d
a	d

,

a	b
b	c
c	c
c	d

2.1 Bottom-up Evaluation

02 Semi-Naive Evaluation

```
edge(a , b).
edge(b , c).
edge(c , c).
edge(c , d).
```

```
path(X , Y) :- edge(X , Y).
path(X , Y) :- path(X , Z), edge(Z , Y).
```

Semi-Naive Evaluation of Program

$\Delta path = edge(X,Y)$
$path = \Delta path$
do
  $\Delta path' = \pi_{X,Y}\big(\Delta path(X,Z) \bowtie edge(Z,Y)\big) - path$
  $path = path \cup \Delta path'$
  $\Delta path = \Delta path'$
while $(\Delta path \neq \emptyset)$
return $path$

2.1 Bottom-up Evaluation

Initialization

$$path^{[0]} = \{(a,b),(b,c),(c,d),(c,c)\}$$

$$\Delta path^{[0]} = \{(a,b),(b,c),(c,d),(c,c)\}$$

Iteration 1:

$$\Delta path'^{[1]} = \{(a,c),(b,d)\}$$

$$path^{[1]} = \{(a,b),(b,c),(c,d),(c,c),(a,c),(b,d)\}$$

$$\Delta path^{[1]} = \{(a,c),(b,d)\}$$

Iteration 2:

$$\Delta path'^{[2]} = \{(a,d)\}$$

$$path^{[2]} = \{(a,b),(b,c),(c,d),(c,c),(a,c),(b,d),(a,d)\}$$

$$\Delta path^{[2]} = \{(a,d)\}$$

Iteration 3:

$$\Delta path'^{[3]} = \emptyset$$

$$path^{[3]} = \{(a,b),(b,c),(c,d),(c,c),(a,c),(b,d),(a,d)\}$$

$$\Delta path^{[3]} = \emptyset$$

path, edge

$I =$

X	Y

,

X	Y
a	b
b	c
c	c
c	d

$T_p^1(I) =$

X	Y
a	b
b	c
c	c
c	d

,

X	Y
a	b
b	c
c	c
c	d

$T_p^2(I) =$

X	Y
a	b
b	c
c	c
c	d
a	c
b	d

,

X	Y
a	b
b	c
c	c
c	d

$T_p^3(I) =$

X	Y
a	b
b	c
c	c
c	d
a	c
b	d
a	d

,

X	Y
a	b
b	c
c	c
c	d

$T_p^4(I) =$

X	Y
a	b
b	c
c	c
c	d
a	c
b	d
a	d

,

X	Y
a	b
b	c
c	c
c	d

2.1 Bottom-up Evaluation

02

Semi-Naive Evaluation

for each IDB p

do $\begin{cases} p^{[0]} := \emptyset \\ \delta(p)^{[0]} := \text{tuples produced by rules using only EDB' s} \end{cases}$

$i := 1$

repeat

$p^{[i]} := p^{[i-1]} \cup \delta(p)^{[i-1]}$

evaluate $\Delta(p)^{[i]}$

$\delta(p)^{[i]} := \Delta(p)^{[i]} - p^{[i]}$

$i := i + 1$

until $(\delta(p)^{[i]} = \emptyset)$ **for each** IDB p

$$\Delta p^{[i]} :- p_1^{[i-1]}, \ldots, \delta(p_j)^{[i-1]}, \ldots, p_{n-1}^{[i-1]}, q_1, \ldots, q_m.$$

$$\delta p^{[i+1]} :- \delta(p_j)^{[i]}, q_1, \ldots, q_m.$$

2.2 Top-down Evaluation

01 **Query-Subquery Evaluation**

edge(a , b).
edge(b , c).
edge(c , c).
edge(c , d).

path(X , Y) :- edge(X , Y).
path(X , Y) :- edge(X , Z), path(Z , Y).
query(Y) :- path(b , Y)

$\{X \mapsto b\}$

$path(b,Y) :- edge(b,Y).$
$path(b,Y) :- edge(b,Z), path(Z,Y).$

$\{Z \mapsto c\}$ $path(c,Y)$

$\{X \mapsto c\}$

$path(c,Y) :- edge(c,Y).$
$path(c,Y) :- edge(c,Z), path(Z,Y).$

- 绑定某些参数的谓词

  P^{γ} (adorned predicate)

- 被绑定的信息向下传递或横向传递到参数中

  用R表示约束关系

  $\{\{X \mapsto b, Z \mapsto c\}, \{X \mapsto b, Z \mapsto a\}\}$

- $< P^{\gamma}, R >$表示一组要回答的子查询

2.2 Top-down Evaluation

02 Adorned predicates 修饰谓词

- P^{γ} 中γ是b和f的序列，称作修饰符，其中的长度正好是P中参数的个数
- 将派生P的规则重写为派生P^{γ}的修饰规则

给一个规则加上修饰符的通用算法如下：
- 如果参数位置是常量，则该参数位置被绑定；
- 如果参数位置是常量，所有出现在规则头的绑定变量在规则体中也是被绑定；
- 如果参数位置是规则体左侧某个原子绑定的变量，则该参数位置被绑定。

$$ar_1 \quad path^{bf}(X,Y) :- edge^{bf}(X,Y).$$
$$ar_2 \quad path^{bf}(X,Y) :- edge^{bf}(X,Z), path^{bf}(Z,Y).$$

2.2 Top-down Evaluation

$ar_1 \quad path^{bf}(X,Y):-edge^{bf}(X,Y).$

$ar_2 \quad path^{bf}(X,Y):-edge^{bf}(X,Z),path^{bf}(Z,Y).$

03 Binding relations 绑定关系

- 通过统一从查询向规则头部自上而下传递的绑定
- 在同一规则主体中从原子向原子横向传递的绑定

■ 使用输入关系来表示自上而下传递的绑定信息

$input\_P^{\gamma}(V) \qquad < P^{\gamma}, input\_P^{\gamma} >$

$input\_path^{bf}(X) \qquad path^{bf}(X,Y)$

■ 使用补充关系来表示横向传递的绑定信息

$sup_j^i(V)$

对于ar_2，有$sup_0^2(X)$，为X提供常量绑定

● $sup_n^i(V)$表示通过计算从规则体构造的所有子查询得到的绑定信息

2.2 Top-down Evaluation

Steps of QSQ Evaluation

Step 4

•规则中的最后一个补充关系用于计算规则头的答案

Step 3

•子查询的产生和计算，以及随后的补充关系的计算

Step 2

•候选规则的第0个补充关系是通过从输入关系中投影出必要的变量来计算

Step 1

•计算从统一查询原子和修饰规则开始

a)设规则i的第j个原子是$P^{\gamma}(V)$，通过将$P^{\gamma}(V)$ 中的变量替换为 $sup_j^i(V')$ 中的常量，构造了来自$< P^{\gamma}, sup_j^i(V') >$的子查询。

b)通过将第j个原子的答案集与$sup_j^i(V')$做join操作，计算出下一个后续的补充关系$sup_{j+1}^i(V')$。

c)由于子查询是由$< P^{\gamma}, sup_j^i(V') >$构造的，因此有必要将绑定信息从$sup_j^i(V')$传递到$input_P^{\gamma}$中，以便可以使用这些相同的求值步骤对这些子查询进行求值。因而从$sup_j^i(V')$中投影出适当的变量来计算$input_P^{\gamma}$。

2.2 Top-down Evaluation

Step 4

•使用步骤3中生成的$ans_path(X,Y)$作为规则头的答案

Step 3

•通过上述的补充关系，可以获得子目标和后续的补充关系

Step 2

•从输入关系中投影出必要的变量，得到补充关系$sup_0^1(X)$和$sup_0^2(X)$，都包含一个绑定关系 $\{\{X \mapsto b\}\}$

Step 1

•将查询与两个候选规则的头部统一，获得输入关系$input_path^{bf}(X)$

绑定信息

$\{\{X \mapsto b\}\}$

a)对于规则ar_1，用$sup_0^1(X)$替换$edge^{bf}(X,Y)$，得到子目标$edge(b,Y)$。规则ar_1的$edge(b,Y)$的答案集就是$\{\{X \mapsto b, Y \mapsto c\}\}$，并直接进入步骤4中$ans_path(X,Y)$的答案集；

对于规则ar_2，同样得到子目标$edge(b,Z)$。$edge(b,Z)$的答案集用于生成下一个补充关系$sup_1^2(Z)$，包含绑定信息 $\{\{Z \mapsto c\}\}$

b)使用$sup_1^2(Z)$和$path^{bf}(X,Y)$来产生下一个子目标：$path(c,Y)$

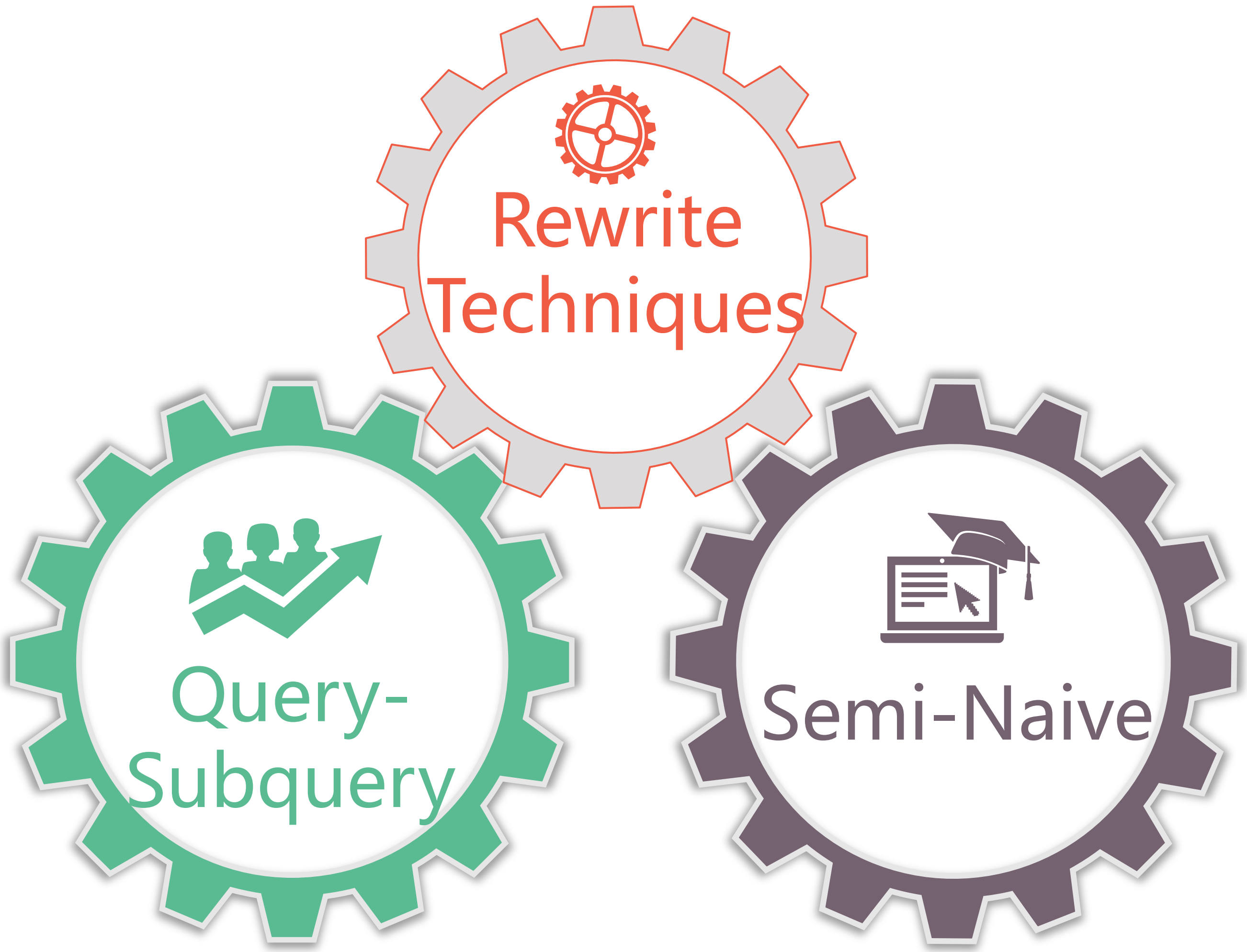
c)上述使用$sup_1^2(Z)$创建了一个新的子目标，需要使用相同的绑定信息来增强$input_path^{bf}(X)$

在这步之后，$input_path^{bf}(X)$包含$\{\{X \mapsto b, X \mapsto c\}\}$

然后返回到步骤2执行相同的步骤来计算子查询$path(c,Y)$

其结果将用于增强步骤4中的$ans_path(X,Y)$

2.3 Magic Sets



$P \xrightarrow{\text{Rewrite}} P'$

- Insight：查询的绑定信息可以表示为谓词，这些谓词插入规则主体
- Magic Sets与QSQ之间有着很强的联系
- 使用自底向上的方法计算这些关系

2.3 Magic Sets

Magic Sets重写的三个步骤：

- 将规则重写为其修饰的版本，其中所需的修饰由查询决定
- 定义绑定谓词和派生到它们的规则
- 通过在规则体中添加绑定谓词重写修饰规则

2.3 Magic Sets

edge(a , b).
edge(b , c).
edge(c , c).
edge(c , d).

path(X , Y) :- edge(X , Y).
path(X , Y) :- edge(X , Z), path(Z , Y).
query(Y) :- path(b , Y)

Adorned rules 修饰规则

一个派生为有n个参数变量谓词的规则有2^n个修饰版本。

$ar_1 \quad path^{bf}(X,Y):-edge^{bf}(X,Y).$

$ar_2 \quad path^{bf}(X,Y):-edge^{bf}(X,Z),path^{bf}(Z,Y).$

Deriving binding predicates 派生绑定谓词

两种类型的关系用于约束此程序的自顶向下计算：
输入关系$input_path^{bf}(X)$和规则ar_i中每个原子j的补充关系$sup_j^i(V)$。

$m_1 \quad input\_path^{bf}(b).$

$m_2 \quad sup_0^1(X):-input\_path^{bf}(X).$

$m_3 \quad sup_0^2(X):-input\_path^{bf}(X).$

$m_4 \quad sup_1^2(X,Z):-sup_0^2(X),edge(X,Z).$

Rewrite rules using binding predicates 使用绑定谓词重写规则

$m_5 \quad path^{bf}(X,Y):-sup_0^1(X),edge(X,Y).$

$m_6 \quad path^{bf}(X,Y):-sup_0^2(X),edge(X,Z),sup_1^2(X,Z),path^{bf}(Z,Y).$

$m_7 \quad input\_path^{bf}(X):-sup_1^2(X,Z).$

感谢聆听

THANK YOU FOR WATCHING